SQL Interview Questions

Friday, 30 January 2026 12:49 PM

1️⃣ **customers**

customer_id INT PRIMARY KEY,
customer_name VARCHAR(100),
email VARCHAR(100),
city VARCHAR(50),
signup_date DATE

2️⃣ **orders**

order_id INT PRIMARY KEY,
customer_id INT,
order_date DATE,
order_status VARCHAR(20),
total_amount DECIMAL(10,2)

3️⃣ **order_items**

item_id INT PRIMARY KEY,
order_id INT,
product_id INT,
quantity INT,
price DECIMAL(10,2)

4️⃣ **products**

product_id INT PRIMARY KEY,
product_name VARCHAR(100),
category VARCHAR(50),
supplier_id INT,
unit_price DECIMAL(10,2),
discontinued BOOLEAN

5️⃣ **suppliers**

supplier_id INT PRIMARY KEY,
supplier_name VARCHAR(100)

supplier_name VARCHAR(100),
city VARCHAR(50)

6️⃣ **employees**

emp_id INT PRIMARY KEY,
emp_name VARCHAR(100),
manager_id INT,
department VARCHAR(50),
salary DECIMAL(10,2),
hire_date DATE

7️⃣ **payments**

payment_id INT PRIMARY KEY,
order_id INT,
payment_date DATE,
payment_mode VARCHAR(20),
payment_amount DECIMAL(10,2)

SQL QUERY PRACTICE QUESTIONS

1. Find customers who have **never placed any order**.
2. Increase the price of all products by **8%** that belong to categories whose **average price is above overall average price**.
3. Display employees who earn **more than their manager**.
4. Find the **second highest order amount** without using LIMIT.
5. List products that were **never sold**.
6. Delete all orders that **have no corresponding payment**.
7. Find customers who placed **more orders than the average number of orders per customer**.
8. Display suppliers who supply **products in more**

than 3 categories.

9. Find orders where the **total of order_items does not match orders.total_amount**.
10. List cities where **total revenue is higher than the average revenue of all cities**.
11. Find employees who were hired **earlier than their manager**.
12. Retrieve products whose price is **greater than ALL products supplied by suppliers from Delhi**.
13. Find customers who placed orders in **every year present in the orders table**.
14. Update salaries by **12%** for employees whose salary is **below department average**.
15. Find the **highest selling product (by quantity)** in each category.
16. Display customers whose **first order value is greater than their last order value**.
17. Identify suppliers who **do not supply any discontinued product**.
18. Find departments where **average salary is greater than company-wide average salary**.
19. Truncate the payments table.
20. Find customers who made **only one order**.
21. Display orders where **order value is greater than average order value of that customer**.
22. Find products whose unit price is **greater than the average price of their category**.
23. Drop the order_items table.
24. Display employees who have **at least 2 direct reportees**.
25. Find customers who have placed orders but **never used UPI payment**.
26. Delete customers who **signed up more than 5 years ago and never ordered**.
27. List suppliers whose **total supplied product value exceeds ₹5,00,000**.
28. Find orders that contain **products from more than**

one category.

29. Display employees who **share the same manager**.
30. Find customers who spent **more than the highest single order value of any other customer**.
31. Update product price to ₹0 for **discontinued products that were never sold**.
32. Find the **third highest salary in each department**.
33. Display customers whose **total payment amount is less than their total order amount**.
34. Find products that were ordered **more times than the average number of orders per product**.
35. List employees who have **no manager and manage at least one employee**.
36. Find customers whose **average order value is higher than the global average order value**.
37. Find orders where **payment was done on a different date than order date**.
38. Display suppliers who supply **only one product**.
39. Find employees whose salary is **greater than ALL employees in 'Sales' department**.
40. Delete orders that belong to customers from cities where **total revenue is below ₹1,00,000**.
41. Find customers who placed **orders in consecutive years**.
42. Display departments where **maximum salary is less than company average salary**.
43. Find products that were **never ordered but are not discontinued**.
44. Find employees who earn the **same salary as someone in another department**.
45. Update order_status to 'Archived' for orders **older than the oldest payment date**.
46. Find customers who have placed **orders for every category available**.
47. Display employees who are **indirectly reporting to a given manager**.
48. Find orders where **number of items is greater than**

average items per order.

49. List customers whose **total spending is higher than their city's average spending**.
50. Find suppliers who supply products that were **never paid for**.
51. Display employees whose salary is **between the department min and max but not equal to either**.
52. Find products whose total sales amount is **greater than average sales amount of all products**.
53. Delete payments where **payment_amount is less than minimum order total**.
54. Find customers who placed **orders on the same date they signed up**.
55. Display managers whose **average team salary is higher than overall employee average salary**.
56. Find orders where **no product price was changed after the order date**.
57. Find customers whose **largest order is smaller than someone else's smallest order**.
58. Display products that were ordered by **customers from more than 2 cities**.
59. Find employees who have **never been managers but earn more than average manager salary**.
60. Update salary to department average for employees earning **below department median salary**.
61. Find customers who placed **more orders than any customer from Mumbai**.
62. Display products whose **supplier city matches customer city of purchase**.
63. Find orders where **payment amount is split across multiple payments**.
64. Find employees who joined **after the average hire date of their department**.
65. Find customers whose **total payment equals total order amount exactly**.
66. Display products whose **price is higher than**

average price of products ordered in last year.

67. Delete suppliers who **do not supply any active product**.
68. Find customers who have ordered **the same product multiple times across different orders**.
69. Display employees who earn **more than the average salary of all other departments combined**.
70. Find customers whose **total spending is higher than the sum of spending of at least two other customers**.

ADVANCED SQL PRACTICE QUESTIONS (SHUFFLED)

1. Find customers whose **total spending is greater than the average total spending of all customers**.
2. Display employees who **earn more than their immediate manager**.
3. Find products that were **never ordered but are not discontinued**.
4. Update salaries by **10%** for employees whose salary is **below the average salary of their department**.
5. Find orders where the **sum of order_items does not match orders.total_amount**.
6. Delete customers who **signed up more than 4 years ago and have never placed any order**.
7. Find suppliers who supply **products in more than two distinct categories**.
8. Display customers who have placed **orders in every year present in the orders table**.
9. Find employees who **do not have a manager but manage at least one employee**

but manage at least one employee.

10. Truncate the payments table.
11. Find customers whose **average order value is greater than the overall average order value**.
12. Display orders where the **payment was made on a later date than the order date**.
13. Find products whose **unit price is higher than the average price of their category**.
14. Find employees who **share the same manager**.
15. Drop the order_items table.
16. Find customers whose **largest order value is less than the smallest order value of any other customer**.
17. Update order status to 'Archived' for orders **older than the earliest payment date**.
18. Find suppliers who **do not supply any discontinued product**.
19. Display orders that contain **products from more than one category**.
20. Find employees whose salary is **greater than all employees in the Sales department**.
21. Delete orders that **do not have any corresponding payment record**.
22. Find customers whose **total payment amount is less than their total order amount**.
23. Display products whose **total sales quantity is greater than the average sales quantity of all products**.
24. Find employees who were **hired earlier than their manager**.
25. Find customers who have placed **only one order**.
26. Update product prices by **5%** for products whose **average selling price is below their unit price**.
27. Find suppliers whose **total supplied product value exceeds the average supplier value**.
28. Display customers whose **total spending is**

28. Display customers whose **total spending is higher than the average spending of customers from their city**.
29. Find employees who **have never been managers but earn more than the average manager salary**.
30. Find orders where the **number of items is greater than the average number of items per order**.

```
INSERT INTO customers VALUES
(1,'Amit Verma','amit@gmail.com','Delhi','2019-01-10'),
(2,'Neha Sharma','neha@gmail.com','Mumbai','2018-03-15'),
(3,'Rohit Mehta','rohit@gmail.com','Delhi','2020-07-20'),
(4,'Pooja Singh','pooja@gmail.com','Bangalore','2017-11-05'),
(5,'Kunal Jain','kunal@gmail.com','Pune','2016-06-18'),
(6,'Sneha Gupta','sneha@gmail.com','Delhi','2021-02-14'),
(7,'Rahul Kapoor','rahul@gmail.com','Mumbai','2019-08-22'),
(8,'Anjali Iyer','anjali@gmail.com','Chennai','2020-01-12'),
(9,'Vikas Malhotra','vikas@gmail.com','Delhi','2015-05-30'),
(10,'Nisha Arora','nisha@gmail.com','Jaipur','2018-09-10'),
(11,'Arjun Patel','arjun@gmail.com','Ahmedabad','2019-12-01'),
(12,'Meera Nair','meera@gmail.com','Kochi','2020-10-21'),
(13,'Saurabh Mishra','saurabh@gmail.com','Bhopal','2016-04-19'),
(14,'Ritika Roy','ritika@gmail.com','Kolkata','2017-07-07'),
(15,'Aditya Kulkarni','aditya@gmail.com','Pune','2019-11-23'),
(16,'Pankaj Yadav','pankaj@gmail.com','Delhi','2021-06-05'),
(17,'Swati Joshi','swati@gmail.com','Indore','2018-01-28'),
(18,'Mohit Bansal','mohit@gmail.com','Gurgaon','2020-03-09'),
(19,'Kavita Shah','kavita@gmail.com','Mumbai','2016-12-15'),
(20,'Manish Rana','manish@gmail.com','Noida','2019-04-11'),
(21,'Ravi Saxena','ravi@gmail.com','Delhi','2017-08-01'),
(22,'Divya Khanna','divya@gmail.com','Chandigarh','2018-06-17'),
(23,'Ashish Tiwari','ashish@gmail.com','Lucknow','2020-09-29'),
(24,'Shalini Goyal','shalini@gmail.com','Agra','2019-02-05'),
(25,'Tarun Sethi','tarun@gmail.com','Faridabad','2016-10-14'),
(26,'Priya Malhotra','priya@gmail.com','Delhi','2021-01-03'),
(27,'Nitin Verma','nitin@gmail.com','Jaipur','2017-05-25'),
(28,'Komal Desai','komal@gmail.com','Surat','2018-11-09'),
(29,'Aakash Choudhary','aakash@gmail.com','Udaipur','2020-12-19'),
(30,'Preeti Sinha','preeti@gmail.com','Patna','2016-02-27');
```

```
INSERT INTO suppliers VALUES
(1,'Alpha Traders','Delhi'),
(2,'Global Supply','Mumbai'),
(3,'TechSource','Bangalore'),
(4,'FreshMart','Pune'),
(5,'Urban Wholesale','Delhi'),
(6,'Prime Distributors','Chennai'),
```

```
(7,'Value Hub','Kolkata'),
(8,'NextGen Supply','Hyderabad'),
(9,'Smart Retail','Noida'),
(10,'Omni Goods','Ahmedabad'),
(11,'Blue Ocean','Surat'),
(12,'NorthStar','Jaipur'),
(13,'QuickShip','Gurgaon'),
(14,'EasyTrade','Indore'),
(15,'MegaStore','Bhopal'),
(16,'SupplyMax','Nagpur'),
(17,'DailyNeeds','Lucknow'),
(18,'CityMart','Patna'),
(19,'EcoSupply','Chandigarh'),
(20,'RetailKing','Agra'),
(21,'TradePoint','Kochi'),
(22,'SuperWholesale','Trivandrum'),
(23,'BulkBuy','Udaipur'),
(24,'FastMove','Faridabad'),
(25,'PrimeStock','Meerut'),
(26,'SmartDeal','Ghaziabad'),
(27,'BudgetMart','Kanpur'),
(28,'ValueTrade','Jabalpur'),
(29,'GoodsLine','Raipur'),
(30,'SupplyChainPro','Ranchi');

INSERT INTO products VALUES
(1,'Laptop','Electronics',3,55000,false),
(2,'Mouse','Electronics',3,500,false),
(3,'Keyboard','Electronics',3,1200,false),
(4,'Smartphone','Electronics',8,30000,false),
(5,'Headphones','Electronics',8,2500,false),
(6,'Office Chair','Furniture',4,7000,false),
(7,'Desk','Furniture',4,12000,false),
(8,'Water Bottle','Home',5,300,false),
(9,'Mixer Grinder','Home',6,4500,false),
(10,'LED TV','Electronics',2,42000,false),
(11,'Refrigerator','Appliances',2,38000,false),
(12,'Washing Machine','Appliances',2,32000,false),
(13,'Sofa','Furniture',7,28000,false),
(14,'Dining Table','Furniture',7,35000,true),
(15,'Bed','Furniture',7,40000,false),
(16,'Shoes','Fashion',9,3000,false),
(17,'Jacket','Fashion',9,4500,false),
(18,'T-Shirt','Fashion',9,1200,false),
(19,'Air Conditioner','Appliances',10,48000,false),
(20,'Microwave','Appliances',10,15000,false),
(21,'Laptop Bag','Accessories',11,1800,false),
(22,'Backpack','Accessories',11,2200,false),
(23,'Watch','Accessories',12,6500,false),
(24,'Sunglasses','Accessories',12,3500,true),
(25,'Tablet','Electronics',13,28000,false),
(26,'Power Bank','Electronics',13,2000,false),
(27,'Iron','Home',14,1800,false),
(28,'Vacuum Cleaner','Home',14,9000,false),
(29,'Smart Speaker','Electronics',15,7500,false),
(30,'Router','Electronics',15,4200,false);

INSERT INTO employees VALUES
(1,'Ramesh',NULL,'Management',120000,'2014-01-01'),
```

(2,'Suresh',1,'Sales',85000,'2016-03-15'),
(3,'Mahesh',1,'IT',90000,'2017-07-21'),
(4,'Neeraj',2,'Sales',55000,'2019-05-10'),
(5,'Aakash',2,'Sales',60000,'2020-08-18'),
(6,'Puneet',3,'IT',70000,'2018-09-25'),
(7,'Rahul',3,'IT',75000,'2021-01-10'),
(8,'Kiran',1,'HR',65000,'2016-11-05'),
(9,'Mehul',8,'HR',48000,'2020-04-14'),
(10,'Sanjay',8,'HR',52000,'2019-06-30'),
(11,'Ankit',2,'Sales',50000,'2022-02-01'),
(12,'Rohit',3,'IT',82000,'2017-12-11'),
(13,'Deepak',1,'Finance',95000,'2015-10-20'),
(14,'Vikas',13,'Finance',65000,'2019-03-08'),
(15,'Nikhil',13,'Finance',72000,'2021-06-17'),
(16,'Amit',1,'Operations',88000,'2016-09-14'),
(17,'Sunil',16,'Operations',56000,'2018-07-07'),
(18,'Pooja',16,'Operations',58000,'2020-01-19'),
(19,'Neha',1,'Legal',92000,'2015-02-25'),
(20,'Kunal',19,'Legal',60000,'2021-11-09'),
(21,'Varun',1,'Admin',70000,'2017-04-16'),
(22,'Ritu',21,'Admin',48000,'2020-12-01'),
(23,'Manoj',1,'Support',65000,'2018-08-13'),
(24,'Shalini',23,'Support',45000,'2021-09-05'),
(25,'Tarun',1,'Support',62000,'2016-06-22'),
(26,'Komal',3,'IT',68000,'2022-05-10'),
(27,'Divya',2,'Sales',53000,'2021-07-18'),
(28,'Anjali',8,'HR',49000,'2022-01-03'),
(29,'Saket',13,'Finance',74000,'2018-10-29'),
(30,'Prakash',16,'Operations',60000,'2019-12-12');

INSERT INTO orders VALUES
(101,1,'2021-01-10','Completed',60000),
(102,2,'2021-02-15','Completed',45000),
(103,3,'2021-03-20','Completed',32000),
(104,4,'2021-04-25','Cancelled',12000),
(105,5,'2021-05-18','Completed',75000),
(106,6,'2021-06-22','Completed',5000),
(107,7,'2021-07-30','Completed',42000),
(108,8,'2021-08-05','Completed',15000),
(109,9,'2021-09-12','Completed',38000),
(110,10,'2021-10-10','Completed',22000),
(111,11,'2021-11-01','Completed',28000),
(112,12,'2021-12-19','Completed',48000),
(113,13,'2022-01-08','Completed',9000),
(114,14,'2022-02-14','Completed',35000),
(115,15,'2022-03-20','Completed',40000),
(116,16,'2022-04-10','Completed',12000),
(117,17,'2022-05-15','Completed',18000),
(118,18,'2022-06-22','Completed',30000),
(119,19,'2022-07-07','Completed',52000),
(120,20,'2022-08-30','Completed',46000),
(121,21,'2022-09-10','Completed',38000),
(122,22,'2022-10-01','Completed',25000),
(123,23,'2022-11-11','Completed',33000),
(124,24,'2022-12-20','Completed',29000),
(125,25,'2023-01-15','Completed',41000),
(126,26,'2023-02-18','Completed',16000),
(127,27,'2023-03-12','Completed',22000),

```
(128,28,'2023-04-25','Completed',36000),
(129,29,'2023-05-19','Completed',48000),
(130,30,'2023-06-30','Completed',51000);



INSERT INTO order_items VALUES
(1,101,1,1,55000),
(2,101,2,2,500),
(3,102,10,1,42000),
(4,103,4,1,30000),
(5,104,7,1,12000),
(6,105,15,1,40000),
(7,105,6,1,7000),
(8,106,8,5,300),
(9,107,10,1,42000),
(10,108,20,1,15000),
(11,109,11,1,38000),
(12,110,22,1,22000),
(13,111,25,1,28000),
(14,112,19,1,48000),
(15,113,27,5,1800),
(16,114,13,1,28000),
(17,115,15,1,40000),
(18,116,21,2,1800),
(19,117,16,2,3000),
(20,118,4,1,30000),
(21,119,29,1,7500),
(22,120,30,1,4200),
(23,121,12,1,32000),
(24,122,18,2,1200),
(25,123,26,2,2000),
(26,124,23,1,6500),
(27,125,9,1,4500),
(28,126,28,1,9000),
(29,127,17,1,4500),
(30,128,5,1,2500),
(31,129,10,1,42000),
(32,130,19,1,48000);


INSERT INTO payments VALUES
(1,101,'2021-01-11','Card',60000),
(2,102,'2021-02-16','UPI',45000),
(3,103,'2021-03-22','Card',32000),
(4,105,'2021-05-20','NetBanking',75000),
(5,106,'2021-06-25','UPI',5000),
(6,107,'2021-07-31','Card',42000),
(7,108,'2021-08-07','UPI',15000),
(8,109,'2021-09-15','Card',38000),
(9,110,'2021-10-12','UPI',22000),
(10,111,'2021-11-03','Card',28000),
(11,112,'2021-12-22','NetBanking',48000),
(12,113,'2022-01-10','UPI',9000),
(13,114,'2022-02-16','Card',35000),
(14,115,'2022-03-22','Card',40000),
(15,116,'2022-04-12','UPI',12000),
(16,117,'2022-05-18','UPI',18000),
(17,118,'2022-06-25','Card',30000),
```